The Road Coloring Problem

Weifu Wang

March 15, 2011

Abstract

The road coloring problem is an interesing problem proposed over 40 years ago by Adler, Goodwyn and Weiss. Hundreds of mathematicians worked on this problem and failed to make the conjecture a theorem. Finally, in 2007, it was proved by a 60-year-old Israeli mathematician, Avraham Trahtman. It turns out the prove was quite simple after Trahtman partitioned the graph into cycles and trees. The details are stated below.

1 Introduction

The road Coloring Problem was posted in the year 1970 by Adler, Goodwyn and Weiss. It was stated explicitly for a strongly connected directed finite graph with constant outdegree of all its vertices where the greatest common divisor (gcd) of lengths of all its cycles is one. Road coloring is an edge coloring problem. It could be phrased in the following way: For a graph who has uniform outdegree and is aperiodic, there should exist a coloring for the edges such that for a certain sequence, regardless of the initial vertex, follwing this sequence of color(or word) will always lead to the same vertex.
In Figure 1 for example, for the vertex marked in yellow, regardless of the initial vertex, traversing all nine edges in the walk blue-red-red–blue-red-red–blue-red-red, will lead to the yellow vertex. Similarly, if one will always terminates at the green vertex by using this walk blue-blue-red–blue-blue-red–blue-blue-red.

For almost 40 years, hundreds of mathematicians have tried to solve this problem, even though progress is made, but the conjecture remains unproven until 2007, an Israel scientist finally came up with the prove and made it a theorem. Avraham Trahtman was a Soviet Union scientist. When he came to Israel, he worked as a security guard before he returned to the academic field.

This theorem is extremely useful in the theory of automaton. When the automaton is running and encounters an error, and if the road coloring conjecture is true, the automaton can always follow a certain sequence and go back to the previous "correct" state, regardless of what error it encountered. The road

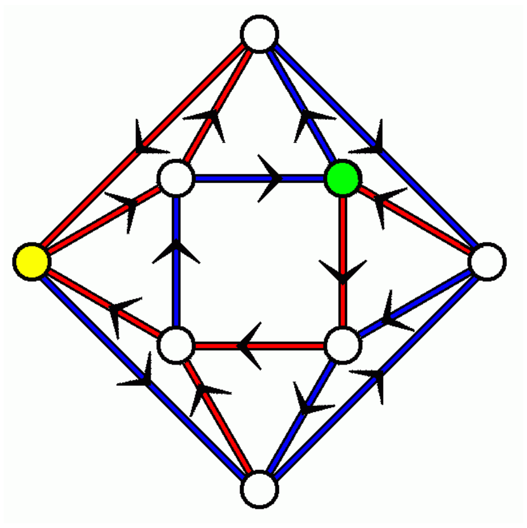

Figure 1: An Example of Road Coloring Problem

coloring is also useful in Symboic Dynamics according to Trahtman.

The idea of the prove is quite simple. Due to the special structure of the graph, we can always partition the graph into cycles and trees. Trahtman proved that there exists such a sub-graph, such that there is only one non-trivial tree outside the cycle with only one leaf with the hightest value. What is more, by his definition of "subgraph" each vertex only has one outgoing edge. Therefore, as long as we get out of the cycle, we can always end up at the same vertex. However, in the cycle, Trahtman proved that one can always follow a sequence and enter the tree regardless of the initial vertex. Then, we can prove this theorem easily.

2 Theorems and Lemmas used in the proof

The following notations and Theorems and Lemmas are from [2].

A digraph D is said to be **out-regular** or have **uniform outdegree** if there is an integer Δ such that od$v = \Delta$ for every vertex v of D. A digraph with uniform outdgree need not have uniform indegree, however.

A digraph D is perodic if it is possible to partition $V(D)$ into $k \geq 2$ subsets $V_1, V_2, ..., V_k, V_{k+1} = 1$ such that if (u, v) is an arc of D, then $u \in V_i, v \in V_{i+1}$ for some i with $1 \leq i \leq k$. Such a partition of $V(D)$ is called a **cyclic k-partition**. Thus D is periodic if there is a cyclic k-partition of $V(D)$ for some integer $k \geq 2$. If D is not periodic, then it is called **aperiodic**. A strong aperiodic graph with uniform outdegree is also called AGW graph.

An arc coloring c of a digraph D is **proper** if every two arcs incident from the same vertex of D are assigned different colors. For a strong digraph D with uniform outdegree Δ a proper $\Delta-$arc coloring c of D is said to be **synchronized** (or **synchronizing**) if for every vertex v of D, there exists a sequence s_v of colors such that for every vertex u of D, the directed walk with initial vertex u determined by s_v has terminal vertex v. In this case, the sequence s_v is called a **synchronized** (or **synchronizing**) **sequence** of the vertex v.

If there exist a path in an automaton from the state $\mathbf{p}$ to the state $\mathbf{q}$ and the edges of the path are consecutively labeled by $\sigma_1, \sigma_2, ..., \sigma_k$, then for $s = \sigma_1\sigma_2...\sigma_k \in \Sigma^+$ let us write $\mathbf{q} = \mathbf{p}s$.

Let Ps be the map of the subset P of the graph by help of $s \in \Sigma^+$ and let Ps^{-1} be the maximal set of states Q such that $Qs \subseteq P$. A word $s \in \Sigma^+$ is called a *synchronizing* word of the automaton with transition graph D if $|Ds| = 1$.

A pair of distinct vertices $\mathbf{p}$, $\mathbf{q}$ will be called *synchronizing* if $\mathbf{p}s = \mathbf{q}s$ for some $s \in \Sigma^+$. In the opposite case, if for any s $\mathbf{p}s \neq \mathbf{q}s$, we call the pair *deadlock*. A synchronizing pair of vertices $\mathbf{p}, \mathbf{q}$ of a graphis called *stable* if for any word u, the pair $\mathbf{p}u, \mathbf{q}u$ is also synchronizing. We call the set of all outgoing edges of a vertex a bunch if all these edges are incoming edges of only one vertex.

Let u be a left eigenvector with positive components having no common divisor of adjacency matrix of a graph with vertices $\mathbf{p}_1, .., \mathbf{p}_n$. The i-th component u_i of the vector u is called *the weight* of the vertex $\mathbf{p}_i$ and denoted by $w\mathbf{p}_i$. The sum of the weights of the vertices from a set D is denoted by $w(D)$ and is called the weight of D. Th subset D of a graph G such that $w(D)$ is maximal and $|Ds| = 1$ for some word $s \in \Sigma^+$, let us call if *F-maximal*.

Theorem 1: *There exists a partition of D of F-maximal sets (of the same weight).*[1]

Theorem 2: *Let us consider a coloring of AGW graph D. Stability of states is a binary relation of the set of states of the obtained automaton; denote this relation by ρ.*

Then ρ is a congruence relation, D/ρ presents an AGW graph and synchronizing coloring of D/ρ implies synchronizing coloring.

Lemma 1: *Let w be the weight of F-maximal set of the AGW graphD via some coloring. Then the size of every F-clique of the coloring is the same and*

[1]The subset G of states of an automaton such that w(D) is maximal and $|Ds| = 1$ for some word s, let us call F-maximal, where Ds be the map of the subset P o states of an automaton by help of s.

equal to $w(D)/w$ (the size of partition of D on F-maximal sets).

Lemma 2: *Let F be F-clique via some coloring of AGW graph D. For any words s the set Fs is also an F-clique and any state [vertex]p belongs to some F-clique.*

Lemma 3: *Let An and B ($|A| > 1$) be distinct F-cliques via some coloring without stable pairs of the AGW graph D. Then $|A|-|A \cap B| = |B|-|A \cap B| > 1$.*

Lemma 4: *Let some vertex of AGW graph D have two incoming bunches. Then any coloring of D has a stable couple.*

Definition: *Let us call a subgraph S of the AGW graph D a spanning subgraph of D if to S belong all vertices of D and exactly one outgoing edge of every vertex.*

A maximal subtree of the spanning subgraph S with root on a cycle from S and having no common edges with cycles from S is called a tree of S. The length of path from a vertex p through the edges of the tree of the spanning set S to the root of the tree is called the level of p in S.

Lemma 5: *Let N be a set of vetices of level n from some tree of the spanning subgraph S of AGW graph D. Then in a coloring of D where all edges of S have the same color α any F-clique F satisfies $|F \cap N| \leq 1$*

Lemma 6: *Let AGW graph have a spanning subgraph R of only disjoint cycles (without trees). Then D also has another spanning subgraph with exactly one vertex of maximal positive level.*

Lemma 7: *Let any vertex of an AGW graph D have no two incoming bunches. Then D has a spanning subgraph such that all its vertices of maximal positive level belong to one non-trivial tree.*

Lemma 8: *Let D be an AGW graph having two cycles C_u and C_v. Suppose that either $C_u \cap C_v = \{p_1\}$ or $C_u \cap C_v = \{p_k, ..., p_1\}$, where all incoming edges of p_i develop a bunch from $p_{i+1} (i < k)$. Let $u \in C_u$ and $v \in C_v$ be the distinct edges of the cycles C_u and C_v leaving $\mathbf{p}_1$. Let R_u be a spanning subgraph with all edges from C_u and C_v except u. The spanning subgraph R_v is obtained from R_u by removing v and add u. Then at least one of two spanning subgraphs R_u, R_v has a unique maximal tree whose root is $\mathbf{p}_1$.*

Theorem 3: *Any AGW graph D has a coloring with stable couples.*

Theorem 4: *Every AGW graph D has synchronizing coloring.*

3 Some thoughts about the theorems and lemmas

The idea of the theorem and the lemmas are quite obvious. As stated above, the author intends to divide the graph into two parts: tree and cycle. According to the definition of the subgraph, there would be only one edge leaving any vertex on the subgraph and any vertex on the cycle would have the same level. At the same time, there would only exist one non-trivial tree with only one highest level leaf. Therefore, if we follow a sequence s, as soon as we leave the cycle, then we always reach the same vertex since there is only one leaf with the highest level. This would be true only if the graph is a subgraph as mentioned in the definition so we would not have multiple paths on a tree. Therefore, lemma 5, 6 and 7 prove this idea. One of the most important ideas in the paper is subsititution.

Consider Lemma 6. If we only have disjoint cycles on a subgraph R, then there must be two edges u and v such that $u = \mathbf{p} \to \mathbf{q}$ and $v = \mathbf{p} \to \mathbf{s}$ and $\mathbf{s} \neq \mathbf{q}$. We can replace v by u, then only $\mathbf{s}$ has the maximal level in the new spanning subgraph. This is an important idea, and it is also used in proving lemma 7. By subsititution, the author proved that there can be only one non-trivial tree with one maximal level if no vertex has two incoming bunches. This leads to the final proof of the Road Coloring Theorem.

Before Trahtman proved this theorem, there were others mathematicians tried to prove this problem and made some progress. Their basic approach was to simplify the graph and partition the graph based on weights.

4 Algorithm for Road Coloring Problem

Soon after the author proved the Road Coloring Theorem, he developed a new algorithm for the road coloring problem.

Even before Trahtman proved the theorem, reserchers have been developing algorithm for this problem. In DNA computing, scholars have developed an algorithm based on the massive parallel computing of sequences of length $O(n^3)$ related to the road coloring problem. In [1], the author developed an algorithm of a $O(n^2)$ complexity in majority cases $O(n^3)$ in worst cases. It is quite a positive result. It is much faster than we would image to calculate a "synchrionizing word" on an AGW graph. As Trahtman points out, we can apply this algorithm in automaton theory, symbolic dynamics, even DNA computing and other fields.

The main steps of the algorithm is built on the theorems and the lemmas stated above. The algorithm can be applied to any graph, no matter whether they are AGW graphs or not, because the algorithm checks the property of the graph before it actually assigns colors.

The main algorithm flow is stated as follows. First, the algorithm checks if the graph has a sink. This step is just to check if the graph has a stable couple or not. If there are multiple sinks, then, lemma 3 is applied to reduce the graph into a smaller graph using the congruence property. This is repeated until there is only one sink in the graph. However, if the graph does not have a sink, then the algorithm returns the error message that the graph is not an AGW graph.

Then, the algorithm use a function "FindLoopColoring" to find the sychronizing words, which is also the coloring sequence. After this, the algorihtm needs to check if the gcd of all the circles are 1 or not. If not, the algorithm terminates. If the gcd is 1, then the algorihtm proceeds.

The following is the main part of the algorithm. Here, for each color we found previously, the algorithm checks for each vertex if it has two incoming bunches, and then finds the spanning subgraph, change it to the subgraph with only circles. Then, the algorithm uses lemma 7 to change it into a graph with only one non-trivial tree with maximal level. At this point, the algorithm has almost finished all the main jobs.

The complexity of the algorithm is usually $O(n^2)$ since the finding subgraph and modify function are linear. Then looping the colors makes the algorithm quadratic. However, the worst case of the algorithm is $O(n^3)$ since if the number of edges in the cycles grows slowly, the size of the graph also decreases slowly. The loops do not appear and the case of two ingoing bunches emerges rarely. This leads to the $O(n^3)$ in worst cases. What is more, the space complexity is also quadratic.

Trahtman implemented this algorihm in the freeware package TESTAS. which can be found at (http:// www.cs.biu.ac.il/ trakht/syn.html).

References

[1] Marie-Pierre Béal and Dominique Perrin. A quadratic algorithm for road coloring. *CoRR*, abs/0803.0726, 2008.

[2] A. N. Trahtman. The road coloring problem. *CoRR*, abs/0709.0099, 2007.